# Technology Readiness Levels and Maturation Approaches for ISHM Technologies

**Ryan Mackey**

***Jet Propulsion Laboratory***

# Talk Outline

- > **Sources of difficulties**
  Why is ISHM behind?

- > **Overview of regular TRLs**
  Are our difficulties unique, or common to all technology efforts?

- > **Previous inadequacies:  What did we do for software?**
  What were the important differences?  Are they similar to the ISHM case?

- > **ISHM specific changes to TRL**
  Proposed new guidelines, as simple and non-invasive as possible

- > **Ways to handle the problems**
  Near-term missions:  Looking ahead five to ten years

- > **Example:  F-18 Experience**
  Applying some of these ideas in a pathfinding experiment

# Difficulties of ISHM

> **ISHM technologies are inherently difficult to mature**
- Treats exceptional behavior by definition
- Poorly understood scenarios, few to no examples
- VERY expensive to test thoroughly
- Difficult to develop without retrofitting

> **Key needed features of ISHM have never been attempted**

> **State-of-art examples do not fully enclose ISHM**
- 777, MER: Complicated system health management, but not truly integrated
  - 777: Driven by maintenance, flight schedules, ground operations
  - MER: Complex reasoning, even on-board behavior modification, but still largely a "learning" system
  - Cassini, other huge robotic spacecraft: Integrated, real-time response, close association with flight controls, but focused on reflexive actions rather than optimizing long-term reliability ("health *maintenance*")
- Missing features: Reasoning, system-level management, integration of different functions, flight- and safety-criticality

# Maturation Hurdles

> Five particular hurdles complicate ISHM maturation

- **Enumerability of states**
  - E.g. preparing for and testing multiple hypotheses
- **The Problem of Nominal**
  - What does "Nominal" mean in a grey-scale universe?
- **Can't Really Test ISHM**
  - Ability to test large components, let alone entire systems to failure, is rare
- **Access to Data**
  - Chicken and egg problem, usually pushes us towards "retrofit"
- **Algorithms vs. Models**
  - Certify a model? Or re-test the whole system just to reconfigure one element of a model? What to do?

# Brief Recap of Ordinary TRL

> **Note focus on hardware**

> **Key points:**

- Performance Model
- Relevant Environment
  - "Stress testing"
- Meaning of Validation
  - Tightly coupled to performance model

### Hardware Systems

| | TRL | |
|---|---|---|
| Basic technology Research | LEVEL 1 | Basic principles observed and reported |
| | LEVEL 2 | Technology concept and/or application formulated |
| Research to Prove Feasibility | LEVEL 3 | Analytical and experimental critical function and/or characteristic proof-of-concept |
| Technology Development | LEVEL 4 | Component and/or breadboard validation in laboratory environment |
| | LEVEL 5 | Component and/or breadboard validation in a relevant environment |
| Technology Demonstration | LEVEL 6 | System/subsystem model or prototype demonstration in a relevant environment (Ground or Space) |
| System or Subsystem Development | LEVEL 7 | System prototype demonstration in a space environment |
| System test, Launch & Operations | LEVEL 8 | Actual system completed and "flight-qualified" through test and demonstration (Ground or Space) |
| | LEVEL 9 | Actual system "flight-proven" through successful mission operations |

> **Other Issues**

- Is "flown in space" equal to "validated in space?"
- When is a validation considered "full-scale?"

# TRL for ~~Software~~ Information Technologies

> **What did we need to change?**

- **What is *our* "relevant environment?"**
  - Most environmental issues have already been addressed by hardware providers
  - IT environment is largely virtual, also much more emphasis on interfaces
- **Execution environment? Development environment? Tools?**
  - Needed to introduce more of the IT development process
  - Unlike hardware, development methods – debugging tools, compilers, etc. – have a complex impact on technology performance beyond feasibility
- **Interfaces**
  - Many more interfaces in information environment
- **Issue of running "on" hardware**
  - Is it possible to mature IT without maturing hardware in parallel?

> **What are the big difficulties facing software?**

- **Left to last minute**
- **Huge state-space**
- **Complex and badly understood environment**

# Suggested TRL Changes for ISHM

- *TRL 1* – Identified/invented and documented a useful ISHM technology with a qualitative estimate of expected benefit.
  - Basic functional relationships of a potential application formulated *and shown to be compatible with reasonable ISHM architectures and testing requirements*.
- *TRL 2* – Completed a breakdown of the ISHM technology into its underlying functions and components, and analyzed requirements and interactions with other systems.
  - Defined and documented requirements for operation, interfaces, and relevant mission phases.
  - Preliminary design assessment confirmed compatibility *with the expected ISHM architecture*.
- *TRL 3* – Major functions of ISHM technology prototyped to prove scientific feasibility. Successful preliminary tests of critical functions demonstrated and documented, leading to a preliminary performance estimate.
  - Experiments with small representative data sets conducted.
  - *Execution environment and development tools required to conduct these tests, such as modeling tools, defined and documented*.

# Suggested TRL Changes for ISHM

> ***TRL 4*** **– Prototype completed on laboratory hardware and tested in a realistic environment simulation.**

- **Experiments conducted with full-scale problems or data sets in a laboratory environment and results of tests documented.**
- ***Development ISHM infrastructure completed as needed for the prototype.***
- **A model of ISHM technology performance, adequate for prediction of performance in the intended application, must be documented as a result of these tests.**

> ***TRL 5*** **– Prototype refined into a system and tested on simulated or flight-equivalent hardware.**

- ***Interaction environment, including interfaces to other systems, defined and included in the testing environment.***
- ***Rigorous stress testing completed in multiple realistic environments and documented.***
- ***Performance of the technology in the relevant environment must be documented and shown to be consistent with its performance model.***

# Suggested TRL Changes for ISHM

- *TRL 6* – **System ported from breadboard hardware testbeds to flight hardware and tested, along with all other needed components, in realistic simulated environment scenarios.**
  - **ISHM technology tested in complete relevant execution environment.**
  - ***Engineering feasibility fully demonstrated.***

---

- **Early TRL: Need understanding of ISHM architecture**
  - **Defines inputs, outputs, timing, and performance requirements**
  - **Integral part of the relevant environment.**
- **Relevant environment:**
  - **Prototype or skeleton ISHM architecture, conforming to the envisioned final application**
  - **Sensors, computing hardware, message passing, etc. are all defined by that architecture**
  - **Stress-testing, for purposes of ISHM, means injection of faults – either simulated or real – that are considered limiting cases, either in terms of sensitivity, timing, or severity.**

# Possible Maturation Solutions

> Design for data capture

- Few programs currently realize the long-term value of system data
- Includes assembly and initial tests, and rarely requires additional sensors
- ISHM can also help find and correct errors during vehicle assembly and test

> Prototype ISHM using mission or vehicle analogues

- Useful surrogate systems may exist
- Higher re-flight rates and greater fault injection and mission variation
- Confront “flight realism” as early as possible

> Revisit flight-critical system requirements

- Not every fault can be anticipated, let alone invoked for testing purposes
- More realistic approaches:
  - Verify models through iterative “pathfinder” techniques
  - Certify resilience in case of “broken” ISHM
  - Separate highly critical, reflexive ISHM and test independently

> Employ model-based engineering (MBE) approaches, autocoding, etc.

- Model-dependence emphasizes need for accuracy, completeness, and interoperability
- Autocoding, automatic model abstraction, and model checking simplifies certification
- Especially valuable as vehicle evolves

> Construct and maintain a centralized “meta-environment” combining system, subsystem, and component models into a unified simulation

- Maintains data and domain knowledge
- Use to test and certify ISHM technologies
- Contains all “Relevant Environments” for different ISHM components

# Mapping Solutions to Problems

| | Design for Data Capture | Analogue Missions | Negotiated Flight-Critical Requirements | Autocoding and MBE | Meta-Environmental Models |
|---|---|---|---|---|---|
| **Difficulty of Full Scale Testing** | Gradual testing as system is assembled | Full-scale tests of equivalent system | More realistic testing requirements | Organize unit testing | Provide end-to-end simulation at various levels of detail |
| **Poor Access to Data** | Capture data and domain knowledge during assembly and test | Access to extensive data of analogous system | | | Playback and integration of captured data, synthesis with simulated data |
| **Imprecise Definition of Nominal** | Mechanism to gather nominal data from system as it is built | Realistic flight testing, including nominal variation | | | Collect known variation within nominal |
| **Algorithms vs. Models** | Test models against actual data as system is constructed | Test ability to generate and include model updates | Permit certification of models without recertifying entire ISHM | Automated generation and checking of models | |
| **State Explosion** | | | Permit separation of ISHM to allow abstraction, independent certification | Allow optimal model testing using exhaustive or branch/bound methods | Provide and maintain a meaningful abstraction of state space |

# Example Maturation Effort: F-18 Testbed

> *Aircraft validation is one example of a "surrogate spacecraft" for NASA ISHM maturation*

- REAL TEST DATA, and lots of it
  - > 40 hours of flight data in under three months
- Simulating challenging situations for ISHM ("breaking stuff") is easier
  - Can include exceptional situations, dummy broken hardware, even real broken hardware
  - Opportunistic scenarios
- Aircraft are in some ways a *better* fit to human-rated flight than robotic spacecraft
  - Crew-rated vehicle
  - Complex environmental interaction
  - Similar time horizons (seconds, not hours)
- Differences between space and aircraft *IT environment* can be closed easily
  - Flight computer, OS, databus, power spec all doable
  - Could include additional space hardware, e.g. sensors, true interfaces
- Cheaper and faster
  - Needed to hit moving targets at NASA

# Maturation Experiment: IMS and BEAM

> IMS: Inductive Monitoring System (ARC)

> BEAM: Beacon-based Exception Analysis for Multimissions (JPL)

- Two separate mid-TRL ISHM technologies
- Monitoring and data interpretation ("Anomaly Detection")
- Potential for real-time, on-board operation

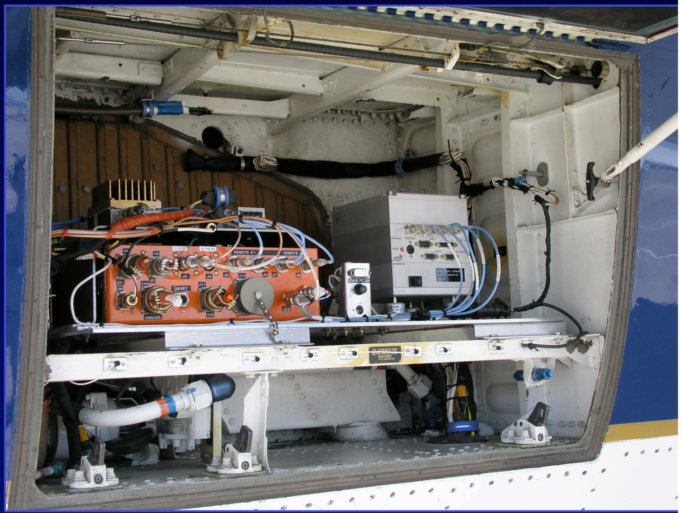
> Experiment Objectives:

- Establish usefulness of F-18 testbed as a technology demonstrator
  - Characterize testbed for developers
  - Generate process of readiness review, integration on aircraft, flight, data recovery, analysis, improvement
  - Understand adaptability of testbed to new hardware
- Deliver hardware and software
- Access to data
  - How long to first flight? Reflight? Quality of data?
  - Other "real world" effects?
- Benchmark F-18 process using BEAM and IMS as first customers

# F-18 Maturation: Where are we today?

> F-18 proved to be an effective "icebreaker"

- Acquired needed hardware
- Achieved first data-taking flight in ~ 4 months
- Introduced realistic flight issues to developers
  - Power
  - Forces ready-for-flight reviews, etc.
  - Computing hardware constraints
  - Computing hardware idiosyncracies
  - Co-development of software and hardware
- Tremendous access to data
  - 23 individual flights, realistic operation
- Ability to modify, retest, refly without lengthy delays
  - In practice, less than one week between update cycles
- Greater access to pilots, aircraft crew chief, etc.
- Flight and simulation both available
  - Also access to existing aircraft models (follow-on work)
- Tested nominal, in-flight faults, plus new discoveries…



# In Conclusion

- ISHM poses special challenges, but nothing intractable
  - Know the common problems and plan to counter them early
- ISHM puts a new spin on the standard TRLs
  - Similar to “hardware” vs. “software” concerns
  - Introduced some ISHM specific changes to TRL, keeping as simple and non-invasive as possible
  - Concepts of “Relevant Environment” and “Performance Model” take on more complex meaning
- Proposed a “covering set” of mitigation approaches
  - Can solve the worst problems by considering requirements, knowledge capture, modeling, and broader applications of MBE
- Example: F-18 Experience
  - Applied four of five mitigating techniques to mature technologies
  - So far, so good